

# Anwendung auf andere CM-Systeme, Vergleich mit anderen Frameworks

Gruppe 5:

- Joomla (Niklas)
- WordPress (Jan)
- Drupal (Jan / Niklas)
- Typo 3 (Ilyas)

Inhalt:

- **Grundlegende Funktionen**
- **Erweiterungstypen**
- **Softwarearchitektur (Pattern)**
- **Datenbankmanagementsystem**
- **Rechtesystem**
- **Allgemeine Betrachtungen**
  - Was haben alle CMS gemeinsam?
  - Wo bestehen Unterschiede?

# Joomla (Niklas)

## Grundlegende Funktionen:

Joomla in der Version 2.5 bietet folgende grundlegenden Funktionen, die direkt nach der Installation für Administratoren zur Verfügung stehen:

- **User Manager**  
Verwaltung von Benutzer des CM-Systems, unterteilt in Guest, Registered, Author, Publisher, Manager, Administrator, Super Administrator.
- **Kategorie Manager**  
Verwaltung von Kategorien innerhalb des CM-Systems
- **Artikel Manager**  
Im Artikelmanager können neue Artikel (Content-Seiten) hinzugefügt, bearbeitet und gelöscht werden.
- **Rechtesystem**  
Joomla bietet ein umfangreiches Rechtesystem, welches mittels einer ACL (Access Control List) realisiert wird. Mit dieser Liste wird festgelegt welcher Benutzer welche Objekte (Dienste oder Dateien) sehen oder arbeiten darf.
- **Menü Manager**  
Dieser Manager dient zum Bearbeiten der Menüs. Hier können einzelne Menüpunkte angelegt werden oder auch hierarchisch angeordnet werden.
- **Sprach Manager**  
Der Sprachmanager unterstützt die Multi-Languagefähigkeit, hier können Sprachen installiert/gelöscht werden oder auch als Standard festgelegt werden.
- **Extension Manager**  
Mittels des Extension-Managers kann sowohl das System gepflegt werden, d.h. das CM-System auf die neueste Version zu bringen, die Datenbankstruktur aktuell zu halten, als auch zusätzliche Erweiterungen zu verwalten.
- **Media Manager**  
Der Media Manager dient zur Verwaltung aller Medien die innerhalb des Systems verwendet werden. Zum Beispiel Bilder, Videos oder andere Medien.

## Versionierung:

### Joomla

Untersucht wurde Joomla in der Version 2.5.11, die Version 2.5.0 hat LTS (Long Time Support) und wird mind. 5 Jahre unterstützt. Die Zahl 11 steht für das letzte Sicherheitsupdate, welches am 27 April 2013 erschien.

## Plattform

Aktuell wird das Framework in der Version 12.1 verwendet. Zu finden ist eine umfangreiche Dokumentation unter <http://api.joomla.org/>.

## Erweiterungstypen:

### Komponenten

Komponenten sind die umfangreichste und komplexeste Art von Erweiterung für das CMS. Die Komponente dient zur Verwaltung der Daten, steuert die Darstellung, bietet Funktionalität und übernimmt Aufgaben, die nicht zu den Grundfunktionen Joomla!-Kern-Anwendung gehören. Üblicherweise werden Komponenten über das Backend gesteuert und können mit Modulen und Plugins zusammenarbeiten. Ebenso verwenden Komponenten üblicherweise MySQL Tabellen.

### Module

Module sind hingegen weniger umfangreich, sind jedoch flexibler und mehrmals im Seitenlayout verwendbar. Module können sowohl eigenständig sein, aber auch mit Komponenten zusammenarbeiten oder Bestandteil dieser sein. Oftmals werden Module verwendet um Inhalt einer Komponente darzustellen, dabei greift das Modul auf die Tabellen einer Komponente in der Datenbank zu. Die Modul Position lassen sich im Backend im Modul-Manager verwalten.

### Plugins

Plugins sind ein Stück Programm-Code, die ausgeführt werden, wenn bestimmte Ereignisse auftreten. Mit ihnen lassen sich Programm-Codes über Ereignisse unterschiedlich steuern. Beispielsweise ist ein Editor ein Plugin, dieser wird nur aufgerufen, wenn das Ereignis `onGetEditorArea` auftritt.

### Templates

Templates sind für die optische Darstellung zuständig, sprich alles, was mit den eigentlichen Grafiken, Formatierung, Anordnung der Modul Positionen und Inhalt zu tun hat. Joomla! verwendet sowohl für die Frontend- als auch für die Backend Gestaltung Templates. Für die Erstellung solcher Templates sind lediglich HTML, PHP, CSS und JavaScript nötig. Ebenso lassen sich die Templates im Backend verwalten und bearbeiten.

### Libraries

Eine Bibliothek in Joomla! ist eine Sammlung von Funktionen, die global innerhalb des CMS zur Verfügung stehen, dass heißt mehrere Komponenten können auf diese Funktionen zugreifen und diese verwenden. Externe Bibliotheken können im Installations-Manager installiert werden.

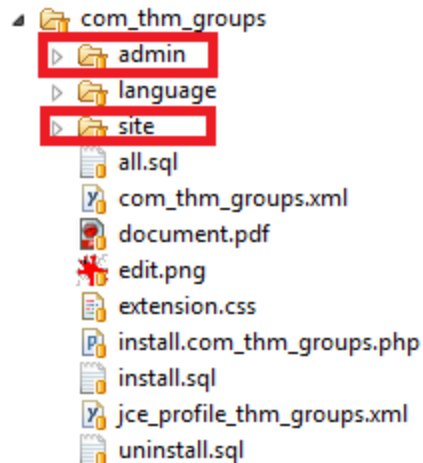
## Softwarearchitektur:

Hier verwendet Joomla das MVC (Model View Controller) Pattern.

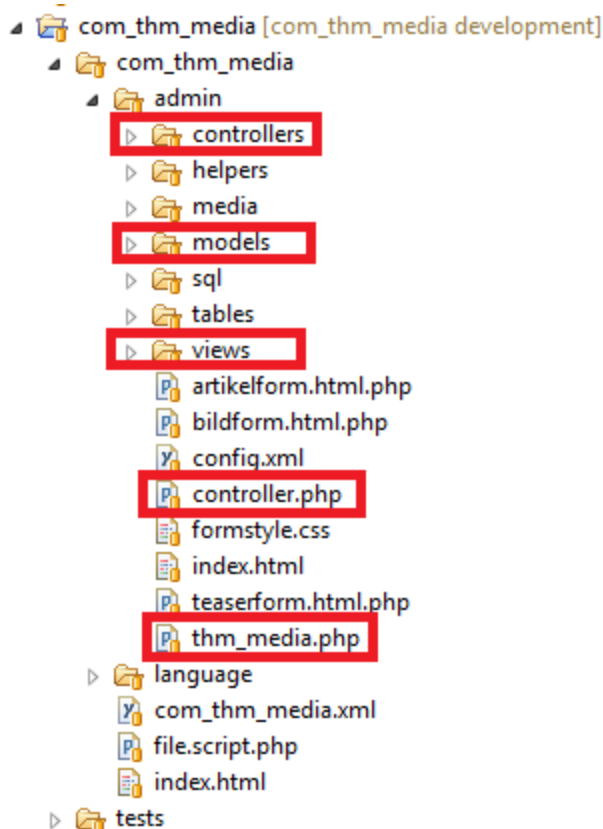
Struktur auf Dateiebene:

### Komponente:

Unterteilung der Quellcodedateien in Front- und Backend.



Dabei sind die Namenskonventionen einzuhalten. Der Root-Ordner enthält einen “admin” und “site” Ordner. Ebenso sind die Setup Datei (XML) und ggf. ein externes Installations-Script (PHP) enthalten.



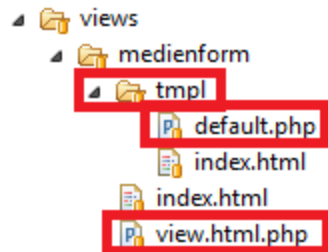
Innerhalb der Unterordner ist das MVC Pattern wieder zu erkennen.

- “models”
- “views”
- “controllers”

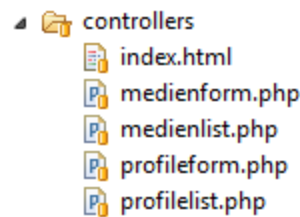
Zusätzlich gibt es einen übergeordneten Controller und eine Einstiegsdatei.

“helpers” enthält Klassen, die von mehreren Sourcefiles verwendet werden.

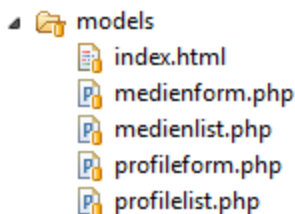
“sql” enthält alle Sourcefiles für die Datenbank, d.h. Installations-Script, Deinstallations-Script und für jede Version eine extra Update-Script.



Views kann mehrere Unterordner enthalten, für jede View einen separaten Ordner. Dieser beinhaltet eine “view.html.php”, welche das Template im “tpl”-Ordner lädt. Hier ist mind immer eine default-Datei enthalten.



Controllers beinhaltet für jede View eine extra Controller-Klasse, hier ist keine Hierarchi erforderlichlich.

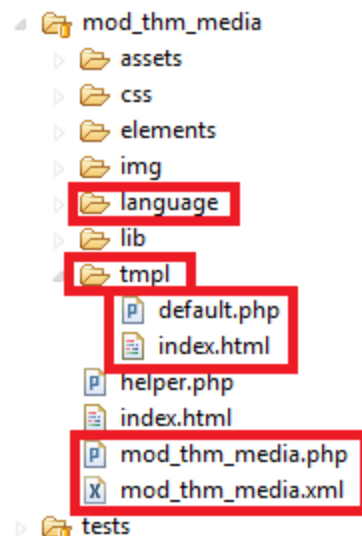


Models enthält für jede View eine Model-Klasse, und lädt die Daten für die jeweilige View.

In jedem Ordner, egal welche Ebene ist eine “index.html” mit leerem Inhalt zu erzeugen.

## Modul:

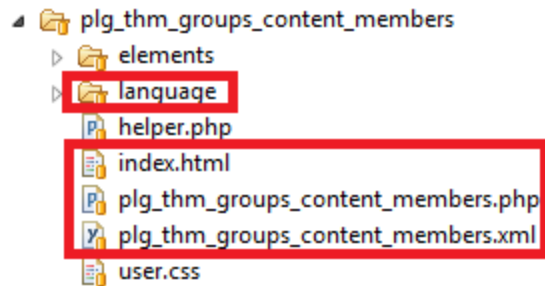
Eine Erweiterung vom Typ Modul beinhaltet nur “Frontend”-Quellcode.



Ein Modul enthält mind. ein Setup-File, eine Einstiegsdatei, einen Sprachordner und mind einen “tpl” Ordner mit einem Default-Template.

Alle anderen Ordner und Dateien sind nicht immer notwendig.

## Plugins:



Ein Plugin beinhaltet ebenso einen Sprachordner, zusätzlich ist ein Setup-File und eine Einstiegsdatei enthalten. Innerhalb der Einstiegsdatei wird das jeweilige Event definiert, wann das Plugin aktiv wird.

## Rechtssystem:

Die Access Control List wurde mit Version 2.5 umgestellt und ist nun umfangreicher. Hier eine kleine Zusammenfassung:

	Version 1.5	Version 2.5
Groups	7 feste Gruppen (Public, Registered, Author, Editor, Publisher, Manager, Administrator, Super Administrator)	Beliebig viele user-definierte Gruppen
User & Groups	Ein Benutzer konnte nur einer Gruppe zugeordnet werden.	Ein Benutzer kann beliebig vielen Gruppen zugeordnet werden.
Access Levels	3 feste Access Levels (Public, Registered, Special)	Beliebig viele user-definierte Access Levels
Access Levels & Groups	Beziehung zwischen Gruppen und Access Levels waren fest.	Gruppen werden ACLs zugeordnet, Jede Kombination ist möglich.

## Datenbank:

Joomla verwendet zur Datenhaltung eine MySQL-Datenbank, dabei kann in der globalen Konfiguration unterschieden werden zwischen MySQL und MySQLi.

Mit PHP5 wurde MySQLi eingeführt. Vorteile sind hier:

- OOP Zugriff, wodurch arbeiten im Team oder an Fremdprojekten vereinfacht wird.
- Größere Geschwindigkeit
- Verbesserte Sicherheit - Verbindung benutzt eine SSH-ähnliche Authentifizierung
- Binärprotokoll wird verwendet, dadurch stehen mehr Funktionen zur Verfügung und die Verarbeitung ist schneller und effizienter.
- Keine Standard Verbindung, Prepared Statements erzwingen eine saubere und somit sichere Programmierung
- Verbesserte Trace- und Debug-Funktionen
- Multi-Queries werden unterstützt.

Jede Erweiterung (üblicherweise nur Komponenten) besitzt seine eigenen Tabellen mit seiner eigenen spezifischen Struktur.

## Links:

Joomla 2.5 Component Tutorial:

[http://docs.joomla.org/Developing\\_a\\_Model-View-Controller\\_Component/2.5/Developing\\_a\\_Basic\\_Component](http://docs.joomla.org/Developing_a_Model-View-Controller_Component/2.5/Developing_a_Basic_Component)

# WordPress (Jan)

## Grundlegende Funktionen

- **Beitrags-Manager**  
Verwaltung von Beiträgen, die auf der Startseite gelistet werden
- **Kategorie-Manager**  
Hier kann man Kategorien erstellen, denen man später Beiträge zuordnen kann.
- **Tag-Manager**  
Verwaltung von Schlagworten
- **Medien-Manager**  
Mit Hilfe der Mediathek kann man Dateien jeglichen Formats bereitstellen, um sie in eine Seite einzubinden oder als Download bereitzustellen.
- **Seiten-Manager**  
Hier werden die einzelnen Seiten der Webpräsenz verwaltet. Jede Seite ist - sofern veröffentlicht - verlinkt über das Hauptmenü.
- **Kommentar-Manager**
- **Design-Manager**  
Bearbeitung folgender Features:
  - Themes: legen das optische Design einer Seite fest
  - Widgets: optionale Bestandteile einer Seite, die nach Belieben platziert werden können (z. B. Kalender, Tag Cloud, ...)
  - individuelle Menüs
  - Kopfzeile
  - Hintergrund
- **Plugin-Manager**  
Erlaubt dem Benutzer, Plugins zu installieren, deinstallieren und direkt zu editieren (PHP-Code).
- **Benutzer-Manager**  
Verwaltung von Benutzerkonten. Ein Benutzer kann folgende Rollen einnehmen:



Administrator, Redakteur, Autor, Mitarbeiter, Abonnent.

- (Werkzeug-Manager)

## Erweiterungstypen

- Plugins
- Werkzeuge  
Spezielle Plugins, die Funktionen für Datenimport und -export bereitstellen. Sie enthalten Schnittstellen zu anderen Blogsystemen, um Datenaustausch und Datenübernahme zu ermöglichen.

## Ordnerstruktur

Es wird kein MVC-Pattern verwendet wie in Joomla.

Unterordner im Stammverzeichnis:

- wp-admin  
Enthält sämtliche PHP-, CSS- und Skript-Dateien für den Backend-Bereich (Administrationsoberfläche).
- wp-include  
Hauptfunktionsordner, der allgemeine Funktionsbibliotheken enthält. Die enthaltenen Funktionen führen z. B. die Datenbankoperationen für das Frontend aus. Der PHP-Code ist hierbei teilweise nach Entitäten in Dateien unterteilt.
- wp-content  
Enthält Code für die Darstellung des Frontend. Des Weiteren sind hier die installierten Plugins (Unterordner "plugins") zu finden.

## **Datenbankstruktur**

Als DBMS wird MySQL verwendet.

Es werden sehr wenige Tabellen verwendet, dadurch bleibt die Struktur relativ übersichtlich. Fast alles, was die Grundlage für die Seiteninhalte bildet (Beiträge, Seiten, Revisionen, ...), ist in der Tabelle "wp\_posts" zu finden. Die Art des Inhalts wird hierbei durch das Feld "post\_type" klassifiziert.

Es existiert eine Tabelle für die Verwaltung der Benutzerkonten ("wp\_user") sowie eine für globale Einstellungen ("wp\_options").

## Typo 3 (Ilyas)

Der Name Typo3 kam durch einen Tippfehler (englisch „typing error“, kurz „typo“) zustande, bei der der TYPO3 Entwickler(Skårhøj) Teile seiner Arbeit verlor. Durch den Erfolg der dritten Version des TYPO, wodurch es populär wurde, wurde „3“ an den Namen angehängt.

Typo3 repräsentiert sich als ein „Enterprise Content Management System“, welches auf PHP basiert.

Die erzeugten Inhalte können zeitgesteuert veröffentlicht werden. Durch eine rudimentäre Workflow-Funktion wird das Projektmanagement unterstützt.

### **Grundlegende Funktionen z.B.**

- Usermanagement

Die Organisation der Benutzer geschieht in Gruppen und in Untergruppen. Jeder Benutzer verfügt nebenbei über einen Workspace, wo er Inhalte vor der Veröffentlichung erstellen und prüfen kann.

- Rechtesystem

Bei Typo3 wird hauptsächlich zwischen Admin und User(Nicht-Admin) unterschieden. Beim Anlegen der Benutzer, bzw. der Benutzergruppen kann je nach Bedarf eine individuelle Rechtevergabe stattfinden. Bei der Rechtevergabe wird zwischen Nutzung der TYPO3 Funktionalität und Zugriffsrechte auf die Verzeichnisse und Dateien unterschieden. Für die Bearbeitung eines Webinhalts benötigt also ein Benutzer sowohl Nutzungsrechte der TYPO3- Funktionalität als auch Zugriffsrechte.

- Sprach Manager

Der Backend verfügt über mehr als 50 Sprachen.

Frontend hingegen unterstützt so viele Sprachen, die für die Lokalisation nötig sind.

- Editoren

- . Typo3 CMS verfügt über einen Rich-Text-Editor zum Erstellen von Webinhalten, wodurch den Benutzern Wissen in HTML-Programmieren erspart bleiben. Die Inhalte auf Front- bzw. Backend können durch Drag & Drop ohne einen großen Aufwand gestaltet werden

## Erweiterungstypen

- Erweiterungen (in TYPO3: Extensions) werden durch Extension-Manager im Backend installiert.

## TypoScript

Wikipedia: „Die Ausgabe von Inhalten im Frontend wird in erster Linie über die Metasprache TypoScript gesteuert. Genau genommen ist TypoScript eine Konfigurationssprache: Was in TypoScript definiert ist, wird in ein systemweites PHP-Array geparkt. Dieses wiederum steuert, welche PHP-Funktionen beim Aufruf der Seite ausgeführt werden. Damit können Eigenschaften und Erweiterungen mit wenigen Handgriffen global für die gesamte Website verwaltet werden. TypoScript ist (bezogen auf die Syntax) objektorientiert.“

## Templates

Die Inhalte werden durch definierte Templates in TypoScript, wie das folgende Beispiel zeigt, dargestellt:

```
page = PAGE
page.10 = TEXT
page.10.value = HELLO WORLD!
page.10.wrap = <h1>|</h1>
```

Ausgabe:

```
<h1>HELLO WORLD!</h1>
```

Die Gestaltung der Inhalte, wie z.B. Navigationsmenüs, kann auch ohne HTML-Templates mit reinem TypoScript-Code erfolgen.

Das Auslesen einer Datenbank kann durch TypoScript stattfinden, ohne PHP zu bedürfen.

## Softwarearchitektur (Pattern)

Model View Controller( MVC): Dieses Pattern kann durch die Integration der entsprechenden Library für die Entwicklung der Extensions zum Einsatz kommen.

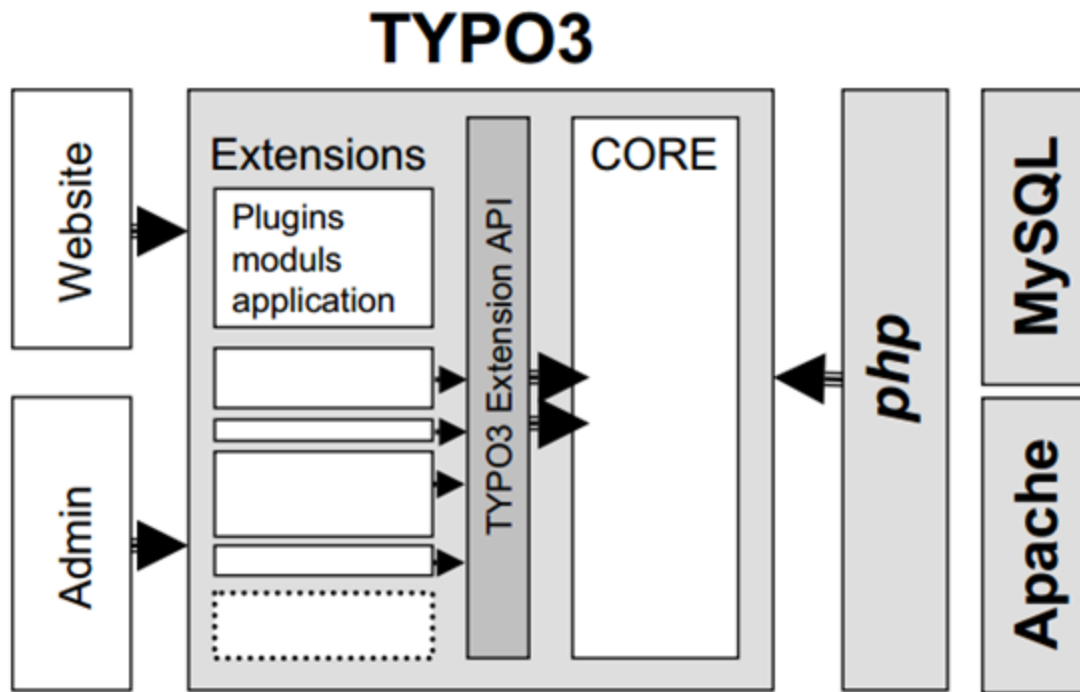


Abb1. Softwarearchitektur

Entnommen

aus: <https://www.dfn.de/fileadmin/3Beratung/Betriebstagungen/bt42/plenumtypo3.pdf>

## Seitenbaum

Die Seiten der Website werden in einer Baumstruktur hierarchisch geordnet dargestellt.

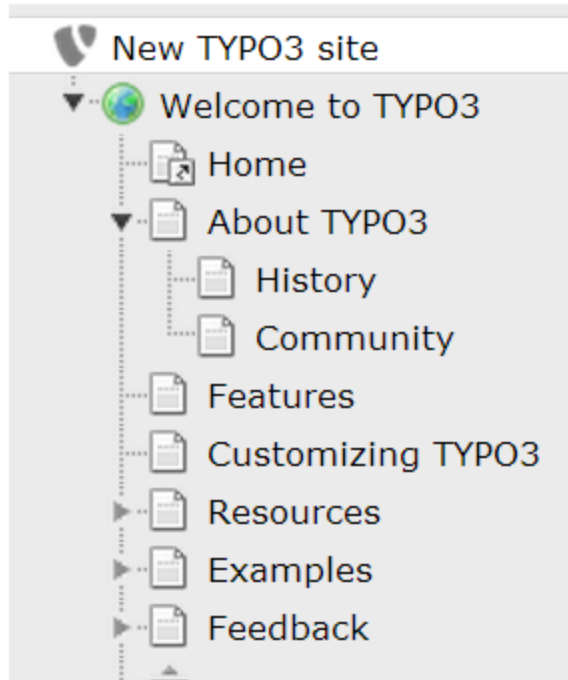


Abb. 02: Seitenbaum

### **Dateibaum**

Dieser stellt die Verzeichnishierarchie und die Ressourcen der Webseite dar.

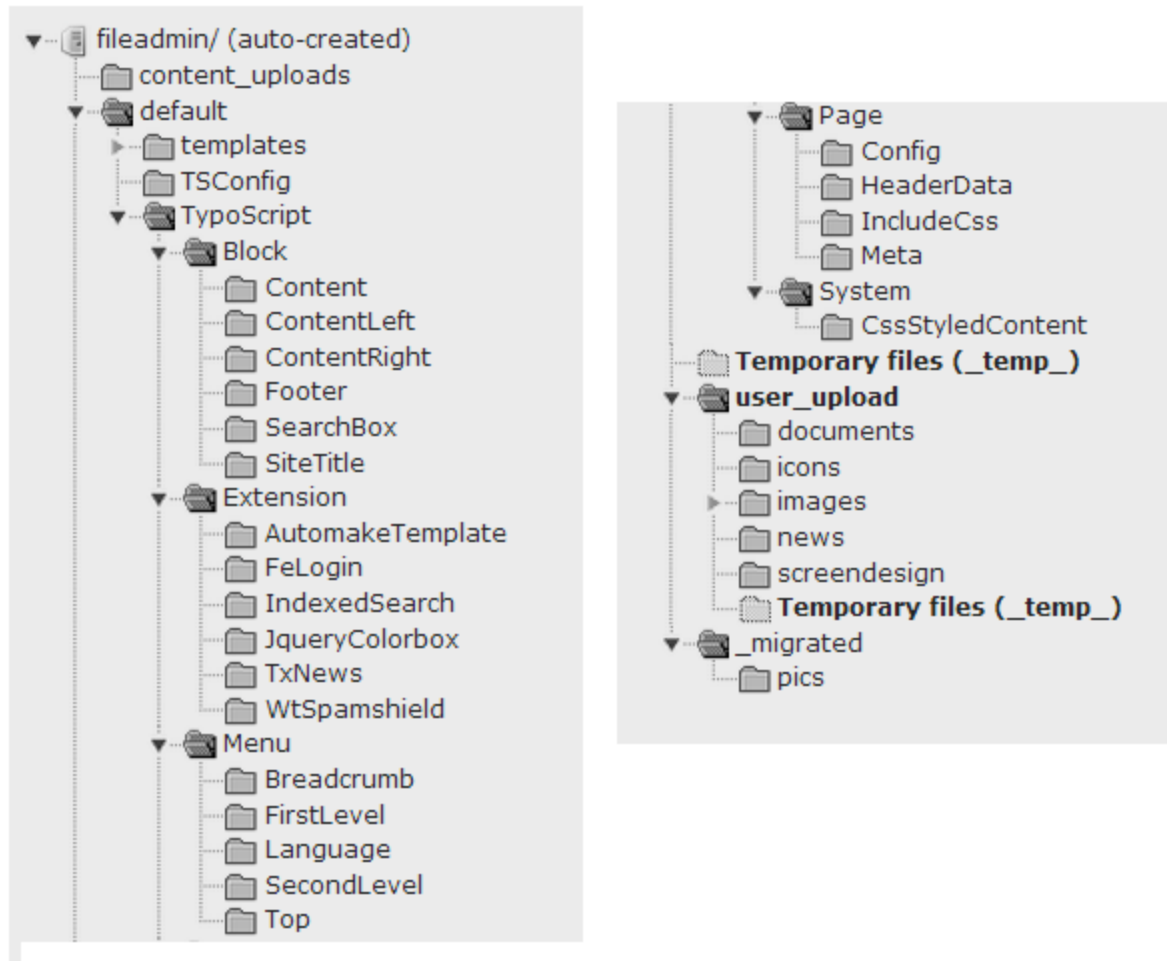


Abb.03: Dateibaum

### Datenbankmodell

-MySQL oder jegliche andere Datenbanksysteme, die von TYPO3 DBA[1]L( z. B Oracle, Postgres etc.) unterstützt werden

---

[1] <http://de.wikipedia.org/wiki/Datenbankabstraktionsschicht>

# Drupal (Jan / Niklas)

## Grundlegende Funktionen

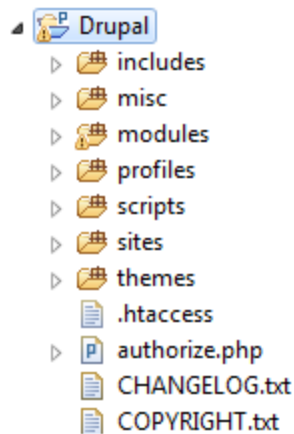
- Verwaltung von "Blocks"
  - Header, Sidebar, Footer, ...
- Verwaltung von Inhaltstypen
  - Artikel
  - "Basic pages" für statische Inhalte
- Verwaltung von Menüs
- Verwaltung von "Taxonomy"
  - Kategorien
  - Tags
- Verwaltung von Themes
  - visuelle Darstellung der Seiten
  - Farben, Anordnung
- Verwaltung von Benutzern
  - Benutzer anlegen und bearbeiten
  - Rechte vergeben
- Verwaltung von Modulen
- globale Einstellungen



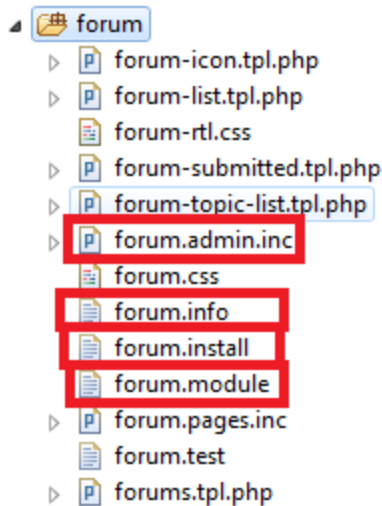
## Softwarearchitektur (Pattern)

- einfache Struktur, daher sehr übersichtlich
- Grundfunktionen in "includes" (\*.inc)
- Themes und deren Templates in "themes"
- Erweiterungen in Form von Modulen
  - zu finden im Verzeichnis "modules"
  - pro Modul genau ein Verzeichnis
  - API-Funktionen und Templates in separate Dateien ausgelagert

## Dateisystem

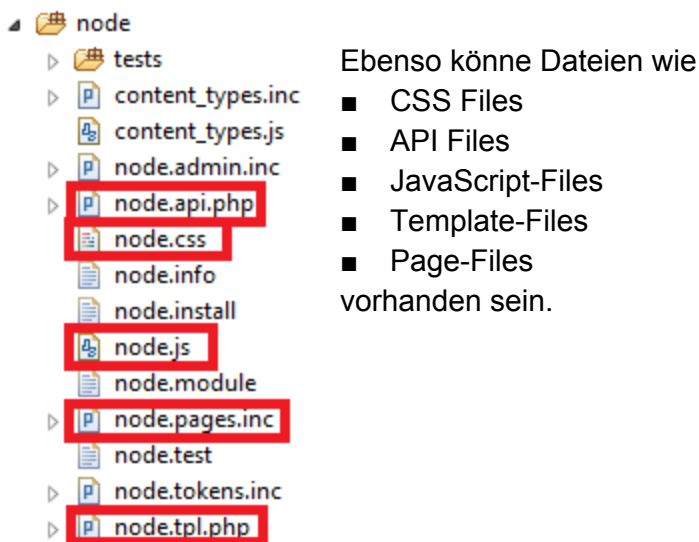


In Drupal werden alle Extensions in den "modules" Ordner installiert. Jeder Erweiterung enthält dort seinen eigenen Ordner. Dieser enthält dann wieder um alle Quelldateien.



Eine Extension enthält immer folgende Dateien, dabei ist die Namensgebung (Naming Conventions) zu beachten.

- extension.admin.inc
- extension.info
- extension.install
- extension.module



Innerhalb der Quelldateien gibt es zahlreiche Funktionen die sich auch an die Namensgebung halten müssen. Hier haben wir die zwingend erforderlichen Dateien untersucht.

- Install File

- forum\_install()
- forum\_enable()
- forum\_uninstall()
- forum\_schema()
- forum\_update\_dependencies()
- forum\_update\_7000()
- forum\_update\_7001()
- forum\_update\_7002()
- forum\_update\_7003()
- forum\_update\_7011()
- forum\_update\_7012()

Hier sind die Standard Funktionen zu erkennen, sowie für jede Version eine update-Funktion incl Versionsnummer.

- Info File

```
forum.info
1 name = Forum
2 description = Provides discussion forums.
3 dependencies[] = taxonomy
4 dependencies[] = comment
5 package = Core
6 version = VERSION
7 core = 7.x
8 files[] = forum.test
9 configure = admin/structure/forum
10 stylesheets[all][] = forum.css
11
12 ; Information added by drupal.org packaging script on 2013-04-03
13 version = "7.22"
14 project = "drupal"
15 timestamp = "1365027012"
16
17
```

Die Info-File enthält alle wichtigen Informationen bzgl. einer Erweiterung.

- **Module-File**  
Enthält alle Funktionen die im Frontend verfügbar sind.
- **Admin-File**  
Enthält alle Funktionen die für einen Administrator zur Verfügung stehen.

## **Datenbanktabelle**

Jeder Erweiterung kann Daten aus beliebigen Datenbanktabellen beziehen. Die Namensgebung hier spielt keine Rolle. Üblicherweise enthält eine Tabelle die speziell zu einer Extentsion gehört aber den Namen.

## **Datenbankmanagementsystem**

- MySQL, PostgreSQL
- verhältnismäßig viele Tabellen
- Größenordnung vergleichbar mit Joomla
- zusätzliche cache-Tabellen für diverse Funktionen

## **Rechtesystem**

- komplexes Rechtesystem
  - Rollen anlegen, Berechtigungen zuweisen
  - Benutzer anlegen, Rollen zuweisen
  - vorgegebene Rollen: anonymer Benutzer, authentisierter Benutzer
  - weitere Rollen können beliebig angelgt werden
- Vergabe der Berechtigungen sehr fein granular möglich
  - geordnet nach Modulen